VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Java herní portál

Java Game Portal

2018 Pavel Kremser

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Zadání bakalářské práce

Student: **Pavel Kremser**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R025 Informatika a výpočetní technika

Téma: Java herní portál
Java Game Portal

Jazyk vypracování: čeština

Zásady pro vypracování:

Cílem je vytvořit internetový herní portál umožňující hraní již vytvořených her v jazyce Java. Součástí portálu je webové rozhraní na platformě JavaEE a také rozhraní pro platformu JavaSE. Každé z těchto rozhraní bude nabízet jinou funkcionalitu.
Systém umožní:
1. Správu uživatelů.
2. Správu dostupných her.
3. Správu odehraných her.
4. Vytváření turnajů (každý s každým, pavouk).
5. Rovněž bude navrženo a použito rozhraní na již existující hry, které bude umožňovat:
a) pouštět hru jako Applet,
b) pouštět hru pomocí technologie Java WebStart,
c) zakládat nové hry,
d) sledovat průběh hry pomocí Java Appletu nebo Java WebStart,
e) získávat průběžné obrázky probíhající hry, bodový stav, pořadí, atd.
6. Připojování hráčů a jejich znovu připojení po ztrátě spojení.
7. Nahrazení hráčů umělou inteligencí po jejich trvalém odpojení (ztrátě spojení), případně znovu napojení hráče, který byl už jednou nahrazen UI a znovu se mu podařilo připojit.
8. Textovou konverzaci hráčů.
9. Naplánování hry na určitý čas.

Práce bude obsahovat:
1. Popis použitých technologií.
2. Implementaci výše popsaného programu.
3. Přizpůsobení a zapojení jedné z existujících her do portálu.
4. Programátorskou dokumentaci řešení s využitím diagramů jazyka UML.

Seznam doporučené odborné literatury:

[1] Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides (Gang of Four): Návrh programů pomocí vzorů. Grada. Praha 2003. ISBN 8024703025
[2] DARWIN, Ian F. Java cookbook. 2nd ed. Sebastopol, CA: O'Reilly, c2004, xxiv, 829 p. ISBN 05-960-0701-9. Dostupné z: http://it-ebooks.info/book/2249/

Dále podle pokynů vedoucího bakalářské práce.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. David Ježek, Ph.D.**

Datum zadání: 01.09.2016

Datum odevzdání: 30.04.2018

doc. Ing. Jan Platoš, Ph.D.
vedoucí katedry

prof. Ing. Pavel Brandštetter, CSc.
děkan fakulty

Prohlášení studenta

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

Dne: 30.4. 2018 .....

Poděkování

Rád bych poděkoval vedoucímu bakalářské práce Ing. Davidu Ježkovi, Ph.D. za pomoc při vytváření této bakalářské práce.

Abstrakt

Bakalářská práce obsahuje herní portál, který je napsaný v jazyce Java. Projekt obsahuje serverovou, klientskou a webovou část. Portál využívá server-klient princip. V prvních kapitolách se popisuje programovací jazyk Java, webové aplikace v Javě a databáze. V další části se popisuje samotná implementace herního portálu. A závěr práce popisuje funkce portálu. Součastí práce jsou také diagramy a zdrojové kódy.

Klíčová slova: Java, herní portál, server-klient, JSF, SQLite

Abstract

This bachelor thesis contains game portal which is written in programming language Java. Project containts server, client and web. Portal is using server to client principle. Programming language Java, web aplication in Java and databse is described in first capitols. In another part is described game portal implementation itself. Functions of portal are described in the end of work. There are also diagrams and source codes as part of the work.

Key Words: Java, game portal, server-client, JSF, SQLite

Obsah

Seznam použitých zkratek a symbolů

JDK	–	Java Development Kit
SDK	–	Software Development Kit
API	–	Application Programming Interface
IP	–	Internet Protocol
GUI	–	Graphical User Interface
HTML	–	HyperText Markup Language
URL	–	Uniform Resource Locator
JSF	–	Java Server Faces
XML	–	Extensible Markup Language
MVC	–	Model, View, Controller
JSP	–	Java Server Pages
JDBC	–	Java Database Connectivity
ODBC	–	Open Database Connectivity
SQL	–	Structured Query Language

1 Úvod

Hraní her je v dnešní době na denním pořádku. Hry hrají všechny věkové kategorie a čím dál větší oblíbenost mají online hry pro více hráčů s nízkou hardwarovou náročností. Proto jsem si vybral bakalářskou práci na téma herní portál. Tento portál umožní jednoduché a rychlé spouštění těchto her.

V této bakalářské práci jsem měl za úkol vytvořit herní portál v programovacím jazyce Java. bakalářská práce bude obsahovat popis programovacího jazyka Java, použité technologie a podrobně popsána implementace herního portálu.

Herní portál je rozdělen na tři programy, které se nazývají Server, Client a Web. Server je základním kamenem celého portálu, zajišťuje veškerou síťovou komunikaci. Client zajišťuje grafické uživatelské rozhraní, které ovládá herní portál. Web umožňuje administraci a zobrazení dat. Portál obsahuje dvě hry na odzkoušení funkčnosti programu.

2 Java

Na vývoji Javy se začalo pracovat v roce 1990, tehdy se jazyku říkalo Oak. Pracovala na něm firma Sun Microsystems, která tenhle jazyk v roce 1995, již pod názvem Java, zveřejnila. V roce 2007 se zveřejnil zdrojové kódy Javy a od té doby je Java vyvíjená jako open source. Od roku 2010 je vlastníkem firma Oracle Corporation.

Java je objektově orientovaný programovací jazyk a jeho syntaxe vychází z jazyka C. Tento programovací jazyk je zcela nezávislý na platformě a má několik edicí, které jsou určeny pro speciální účely:

- Java Card - Java pro čipové karty
- Java ME (MicroEdition) - Java pro mobilní zařízení
- Java SE (Standard Edition) - Java pro osobní počítače
- Java EE (EnterpriseEdition) - Java pro podnikatelské aplikace

2.1 Vývojové prostředí

Vývojové prostředí je software, který usnadňuje práci programátorům. Obsahuje editor zdrojového kódu, kompilátor, debugger a další.

Javu podporuje několik vývojových prostředí mezi ty nejznámější patří Eclipse, Netbeans, BlueJ a JDeveloper.

2.1.1 Netbeans

Netbeans je zdarma. Vlastní ho firma Oracle Corporation. Vzniknul v roce 1996. Primárním programovacím jazykem je Java, ale podporuje i jiné například PHP, HTML, CSS, JavaScript, C/C++ a další.

Nejnovější verzí je NetBeans IDE 8.2.

2.1.2 Eclipse

Eclipse je open source. Vlastní ho firma Eclipse Foundation. Hlavním jazykem je Java, která však může být pomocí pluginů rozšířená o další jazyky jako je třeba C++ nebo PHP. Eclipse neobsahuje většinu nástrojů jako například grafické rozhraní a proto musíme tyto nastroje dodat pomocí pluginů.

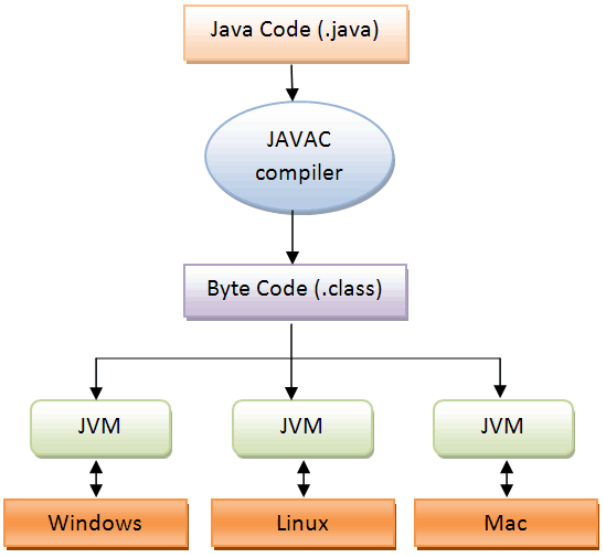
2.2 Java Development Kit

JDK je rozšíření SDK a obsahuje základní nástroje pro vývoj programů v Javě. Součásti JDK jsou:

- Java Runtime Environment
  - virtuální stroje Javy
  - API
- java - zavaděč pro aplikace Java
- javac - překladač zdrojového kódu v jazyce Java do bajtkódu
- jdb - debugger pro ladění programů.
- a další[1]

2.3 Virtuální stroj Java

Virtualní stroj Java pracuje s bajtkódem, který je získáván ze strojového kódu. Virtuální stroj pracuje na různých platformách a díky tomu můžeme jeden bajtkód spouštět na různých zařízeních. "Write once, run anywhere (WORA)"[2]



Obrázek 1: kompilace kódu[3]

2.4 Objektově orientované programování

Objektově orientované programování je programovací paradigma. Není to jen o psaní kódu jiným způsobem, ale musíme i uvažovat jiným způsobem. V objektově orientovaném programování se snažíme vytvářet samostatné a znovupoužitelné objekty.

Všechny programy v Javě jsou objektově-orientované a základním kamenem jsou třídy. Třídy mohou obsahovat:

- proměnné ve kterých je uložen stav objektu
- konstanty
- metody které mění stav objektu a popisují schopnosti objektů

Velice důležité vlastnosti objektově orientovaného programování jsou zapouzdření, polymorfizmus a dědičnost.

Zapouzdření umožňuje skrýt některé metody a atributy tak, aby zůstaly použitelné jen pro třídu zevnitř. Přes rozhraní jí předáváme data ke zpracování.[4]

Dědičnost slouží k tvoření nových datových struktur na základě starých. Kdy nové datové struktury zdědí funkce a vlastnosti po staré struktuře a můžeme přidat ještě další vlastnosti a funkce.

Polymorfismus umožňuje používat jednotné rozhraní pro práci s různými typy objektů. Více objektů může dědit ze stejného rozhraní, ale můžeme si přepsat metody. Takže i přes stejný název a parametry můžou dělat něco jiného.

2.5 Prvky jazyka Java

2.5.1 Síťové spojení

Pro navázání síťového spojení se v javě využívá Socket, který se nachází v třídě java.net.Socket.

Příklad navázání komunikace je, že se na obou koncích komunikace vytvoří Socket. Na straně serveru se vytvoří ServerSocket, který bude znát číslo portu a bude poslouchat. Na straně druhé se Socket pomocí portu a IP adresy pokusí připojit na server. Při úspěšném navázaní spojení se zahájí síťová komunikace, která probíhá pomocí proudů.

2.5.2 Vlákna

Vlákna se využívají k tomu, aby program mohl dělat dvě a nebo více věcí současně. Každá část, která běží samostatně se nazývá vlákno. Takových vláken může mít program neomezené množství. Existují dva způsoby k vytvoření vlákna.

- Poděděním z třídy Thread
- Implementací rozhraní Runnable

Oba dva způsoby využívají baliček java.lang a v obou případech se přepisuje metoda run(), která se spouští metodou start().

```
public class Main {
   public static void main(String[] args) {
      Thread t1 = new MyThread();
      Thread t2 = new MyThread();
      t1.start();
      t2.start();
   }
}
class MyThread extends Thread{
   @Override
   public void run() {
      for(int i = 0; i < 5; i++){
         System.out.print(i);
      }
   }
}
```

Výpis 1: Vytvoření a spuštění vláken

Výstup programu by vypadal asi takhle "0011223344".

2.5.3 Grafické uživatelské rozhraní

Grafické uživatelské rozhraní GUI poskytuje uživateli možnost interaktivního ovládání počítače. Za pomocí vstupních zařízení, jako jsou myš, klávesnice či touchpad, ovládá uživatel grafické prvky na monitoru počítače a tím přenáší své myšlenky a pokyny k akci PC.[6]

Pro grafické rozhraní v Javě využíváme baliček javax.swing. Zobrazované prvky nazýváme komponenty. Mezi takové prvky patří například tlačítko, textové pole a další. Pro vytvoření nových prvků se využívá vytváření instanci. Například pro vytvoření tlačítka budeme vytvářet instanci z JButton.

2.5.4 Výjimky

Výjimka je chyba, k níž dojde za běhu programu. Pomocí subsystému Javy pro zpracování výjimek můžete strukturovaným a řízeným způsobem ošetřit chyby vzniklé za běhu programu.[5]

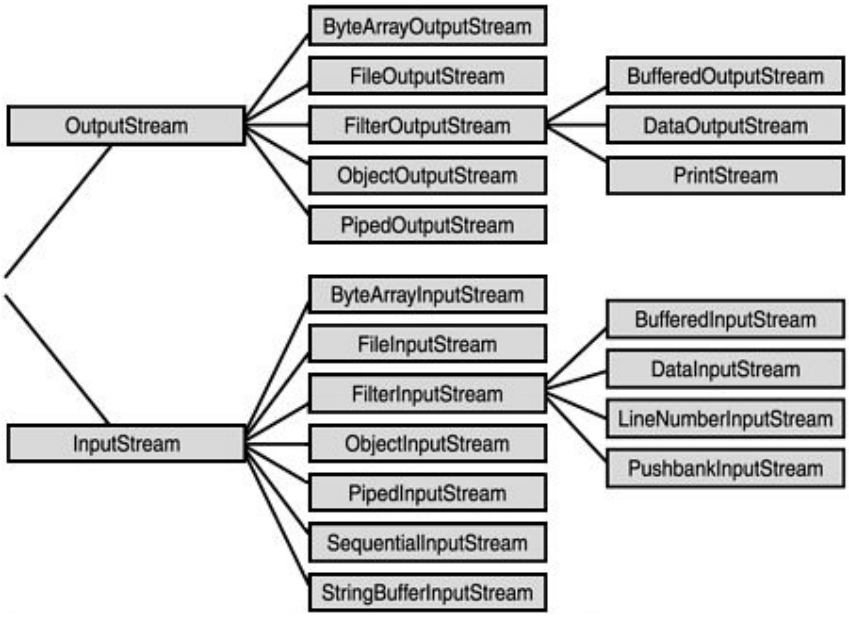
Výjimky v Javě jsou objekty, které jsou odvozeny ze třídy Throwable. Tato třída má dvě podtřídy java.lang.Error a java.lang.Exception.

Výjimky třídy Error představují vážné chyby programu a většinou končí ukončením celého programu.

Výjimky třídy Exception se dají ošetřit a program může pracovat dále. Výjimky se dají ošetřit pomocí konstrukce try-catch-finally. V bloku try zachytíme kus kódu, který je rizikový a je pravděpodobné, že se v něm může vyskytnout výjimka. V bloku catch výjimku zachytíme a zpracujeme. Blok finally je nepovinný a zde můžeme dát kód, který se má provést vždy, takže i v případě kdy byla výjimka zachycena, ale i v případě kdy zachycena nebyla.

2.5.5 Proudy

Proudy se používají pro přenos dat a jsou definovány svým vstupem a výstupem. Vstupy a výstupy mohou být různé zařízení, soubory nebo jiné programy. V javě se proudy dělí na několik typů. Seznam těchto typů můžeme vidět na obrázku 2. Většina tříd pracující s proudy pochází z balíčku java.io.



Obrázek 2: podtřídy InputStream a OutputStream[7]

3 Webové aplikace

Webové aplikace jsou aplikace, kde uživatelským rozhraním je webový prohlížeč. Mají tři obrovské výhody oproti ostatním typům aplikací (desktopové, mobilní, příkazový řádek):

- uživatel nemusí nic instalovat
- vývojář se stará o jen jednu verzi aplikace
- u správně napsaných aplikací existují adresy URL do částí aplikace, např. lze někomu jinému poslat odkaz na stránku s popisem určitého zboží

Webové aplikace se často používají jako uživatelské rozhraní ve vícevrstvých aplikacích.[8]

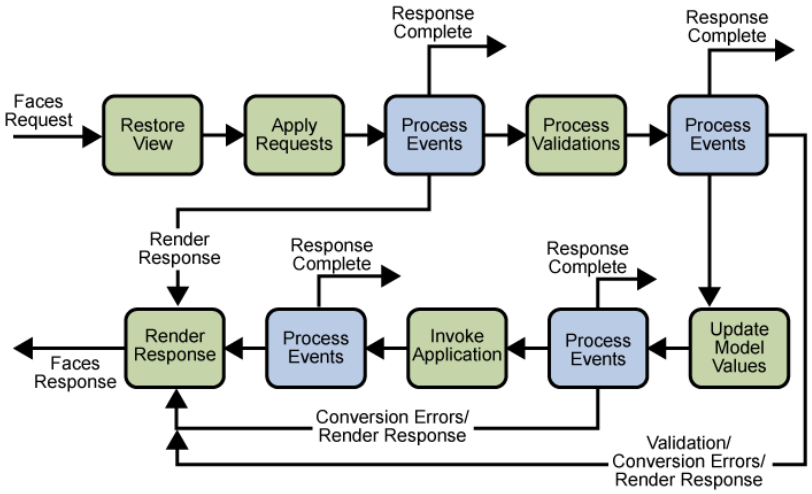
3.1 Java Server faces (JSF)

Tato technologie se začala vyvíjet ve společnosti Sun Microsystems a později vývoj převzala společnost Oracle Corporation. JSF je framework, který pomáhá při vývoji webových stránek v jazyce Java. Uživatelské rozhraní je určeno pomocí speciálních XML tagů a data jsou editována pomocí Java Beanu.

JSF je založen na MVC konceptu:

- Model - Obsahuje informace a data se kterýma aplikace pracuje
- View - Zobrazuje informace pro uživatele z vrstvy Model
- Controller - Reaguje na události od uživatele a provádí změny ve vrstvách Model a View

Celý životní cyklus je rozdělen do šesti fází, kterými se mezi vznikem požadavku a zasláním odpovědi prochází. Životní cyklus nemusí proběhnout celý protože během fází probíhá volání různých metod, které můžou cyklus ukončit nebo jej zkrátit.[10]



Obrázek 3: životní cyklus již načtené stránky[10]

3.2 Java Server Pages (JSP)

Java Server Pages pomocí kombinace Javy a HTML vytváří dynamické weby. Tyto soubory mají příponu .jsp. Volání JSP je jen spuštění metody service(), ve které je ukládán Java kód. HTML kód upravuje vzhled samotné webové aplikace a java kód dynamicky upravuje hodnoty na stránce.HTML kód je ohraničován třemi základními syntaxemi.

- Scriptlet tag <% %> vložení Java kódu do metody _jspService()
- Expression tag <%= %> výpis proměnné nebo řetězcového výrazu
- Declaration tag <%! %> definování tříd, metod a proměnných mimo metodu _jspService() a proto nezabírají paměť při každém načtení stránky

3.3 Aplikační server GlassFish

Server GlassFish začal vznikat v roce 2005 a v květnu roku 2006 byla vydána první verze. Server má hodně společných vlastnosti se serverem Apache TomCat. GlassFish je open source a slouží pro platformu Java EE. Byl vyvinut společnosti Sun Microsystems později přešel vývoj pod společnost Oracle Corporation.

Jedná se o multiplatoformní aplikaci a podporuje Servlety, EnterpriseJavaBeany, Java Persistence API, JavaServerFaces, JavaServerPages a další.

4 Databáze

Databáze je velké skladiště faktů navržené takovým způsobem, aby bylo snadné zpracování těchto skutečností na informace. [11]

Databáze je souhrn strukturovaných dat, které můžeme pomocí databázového systému upravovat, mazat,vyhledávat, porovnávat apod.

4.1 Relační databáze

Relační databáze je velmi využívaná v aplikacích. Nejznámější databázové systémy jsou MySQL, Oracle, MS SQL a další.

Základním prvkem relační databáze jsou tabulky neboli entity, které mají mezi sebou logickou vazbu(relace).

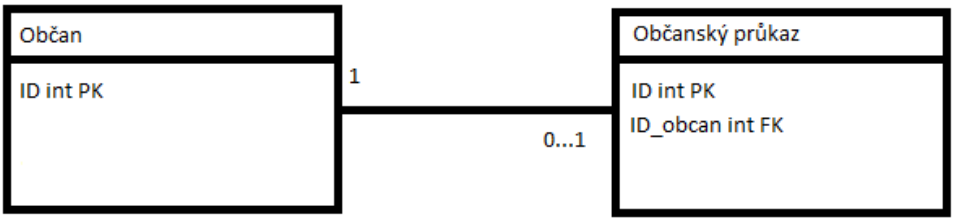
Každá tabulka má řádky a sloupce. Řádkům říkáme záznamy a sloupcům atributy. Sloupce musí mít doménu. Doména je rozsah hodnot, které můžou být v atributu. Z domény si pak databázový systém získá datový typ. Každý sloupec určuje vlastnost záznamů. Každý záznam může mít u jednoho atributu jen jednu hodnotu. V případě že chceme hodnot zadat víc tak si musíme vytvořit novou tabulku, kde budeme odkazovat na záznam, kde tyhle hodnoty patří.

Primární klič je atribut který je v celé tabulce jedinečný a díky němu se můžeme odkazovat na daný řádek v tabulce. Primární klič se může skládat i z více atributů, ale nejčastěji se používá číselná hodnota, která se zvětšuje a nazýváme ji ID.

Cizí klič odkazuje na primární klič jiné tabulky. Pomocí tohoto klíče se určují vazby mezi tabulkami.

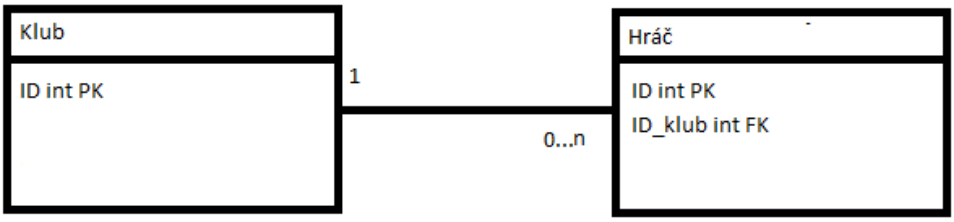
Existují 3 základní typy vazeb a rozdělují se podle multiplicity. Multiplicita určuje, kolik záznamů v první tabulce může být spojeno s jedním záznamem v tabulce druhé a naopak.[12]

Vazba 1:1 je vazba, kde jednomu záznamu odpovídá právě jeden záznam v jiné tabulce. Příkladem může být občan a občanský průkaz. Každý člověk může vlastnit jen jeden nebo žádný platný občanský průkaz.



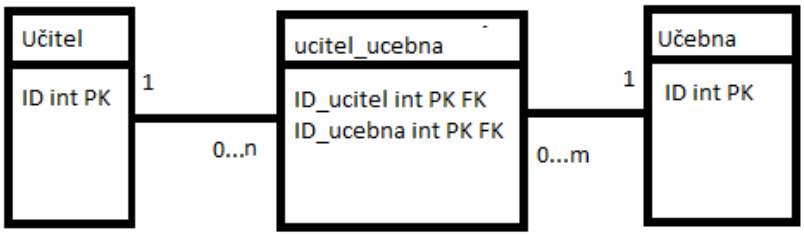
Obrázek 4: příklad 1:1 vazby

Vazba 1:n je vazba, která na jedné straně má jeden záznam a na druhé straně může být záznamů mnoho. Jako příklad může být třeba fotbalový klub a hráči. Jeden klub může vlastnit mnoho hráčů, ale hráč může patřit jen jednomu klubu.



Obrázek 5: příklad 1:n vazby

Vazba m:n je vazba, kde na obou stranách vazby může být mnoho záznamů. Taková vazba nejde vyjádřit jen pomocí cizího klíče, ale potřebujeme vytvořit novou spojovací tabulku. Spojovací tabulka musí obsahovat minimálně dva atributy a to budou cizí a zároveň primární klíče. Příkladem může být učitelé a učebny. Jeden učitel může učit ve více učebnách, ale zároveň v jedné učebně se za celý den může vystřídat více učitelů.



Obrázek 6: příklad m:n vazby

4.2 SQLite

SQLite je relační databázový systém obsažený v malé knihovně. Samotná databáze je uložena v souboru na disku. Pro spuštění SQLite není potřeba téměř žádné konfigurace, ale pokud je nějaká konfigurace potřeba tak ji lze provést pomocí příkazu PRAGMA. SQLite je multiplatformní a podporuje většinu programovacích jazyků.

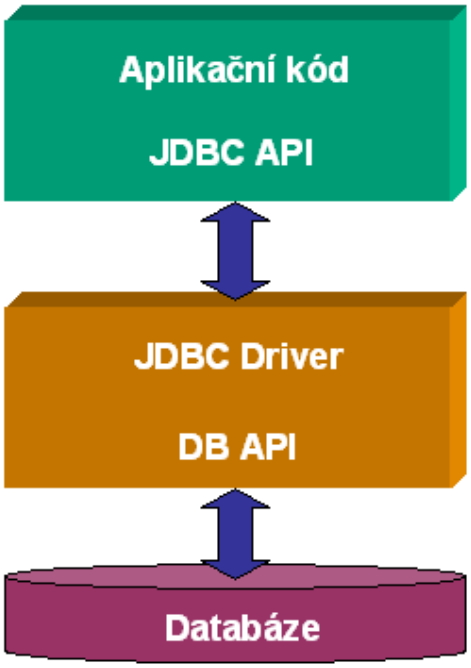
SQLite je pod public domain licencí a vytvořil ho D. Richard Hipp. Díky těmto licenčním podmínkám je tenhle databázový systém hodně rozšířený. Systém je využívány například v Google Chrome, Safari, Android Browser, Skype atd.

Velká nevýhoda SQLite je jeho výkon. Každý insert v SQLite je chápán jako transakce a proto je insert dokončen až tehdy, když přijde potvrzení, že jsou data uložena. Tohle zapříčiňuje nízký výkon systému. Jeho druhou nevýhodou je , že není možnost přístupu k datům během zápisu.

Výhody SQLite je malá velikost a jednoduché použití. Další výhoda je možnost čtení z několika instancí najedou.

4.3 Java Database Connectivity

JDBC je API, které poskytuje jednotné rozhraní k relačním databázím. Pro práci s JDBC je potřeba ovladač používané databáze, ale naprogramovaný kód v javě bude stejný bez ohledu na použitou databázi.



Obrázek 7: zjednodušenné schéma JDBC[15]

```
Class.forName("org.sqlite.JDBC");
Connection c = DriverManager.getConnection("jdbc:sqlite:db.sqlite");
```

Výpis 2: Ukázka vytvoření připojení k SQLite databázi pomocí JDBC

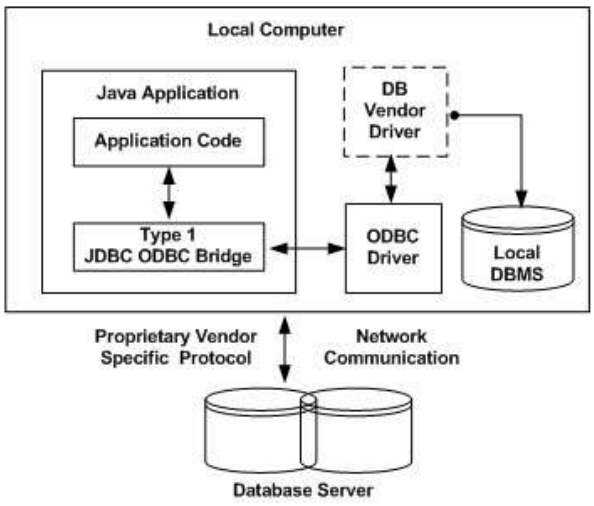
```
stmt = c.createStatement();
ResultSet rs = stmt.executeQuery( "SELECT * FROM user;" );
while ( rs.next() ) {
   id = rs.getInt("id");
   login = rs.getString("login");
   password = rs.getString("password");
   users.add(new UserObject(id,login, password));

}
rs.close();
stmt.close();
```

Výpis 3: Ukázka dotazu provedeného nad databází

Společnost Sun Microsystems vydala JDBC poprvé v roce 1997 jako součást JDK 1.1. Všechny třídy potřebné pro JDBC jsou obsažené v balíčcích java.sql a javax.sql. JDBC ovladače se dělí na 4 typy.

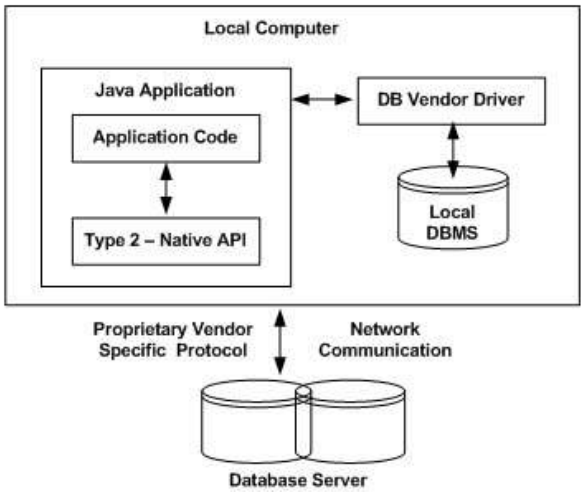
První typ využívá ODBC a přistupuje přes JDBC-ODBC most. Pro tento typ je nutné nainstalovat a nastavit ODBC ovladače na každém stroji, který bude spojení využívat. Vzhledem k tomu, že jsou potřeba dva ovladače, tak výkon bude nižší.



Obrázek 8: první typ JDBC architektury[16]

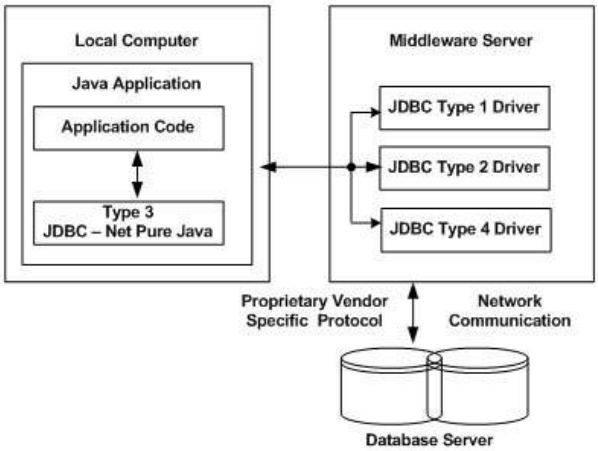
Ovladač typu 2 má za úkol překládat požadavky z JDBC do určitého specifického ovladače, který je v nativní podobě nainstalovaný na počítači a který je určen právě pro jeden typ databáze.[15]

Nevýhody jsou stejné jako u prvního typu, že je třeba specifický software, který musí být na připojeném stroji a využívají se dva ovladače.



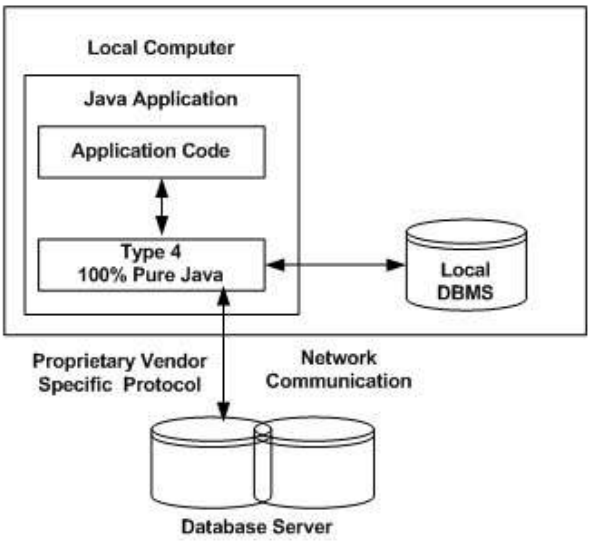
Obrázek 9: druhý typ JDBC architektury[16]

Třetí typ ovladače využívá pouze Javu a aplikační server. Dotazy se přeposílají na server, kde serverová část ovladače přeloží požadavky na konkrétní typ databáze. Tyto přeložené dotazy přepošle do databáze. Výhodou je, že na stroji není potřeba žádný software a při změně databáze stačí jen upravit konfiguraci na serveru.



Obrázek 10: třetí typ JDBC architektury[16]

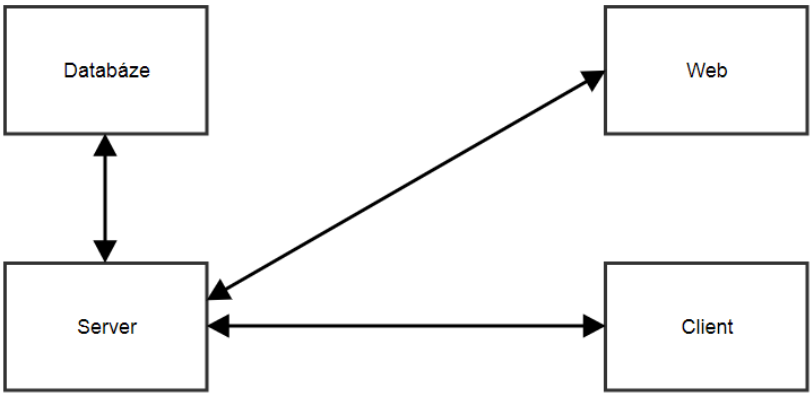
Ovladač typu 4 je napsán celý čistě v Javě s plnou podporou pro optimalizace vzhledem k dané databázi. Výhodou tohoto ovladače je, že klient nemusí být jakkoli konfigurován a nejsou nutné žádné lokální klientské instalace ovladačů.[15]



Obrázek 11: čtvrtý typ JDBC architektury[16]

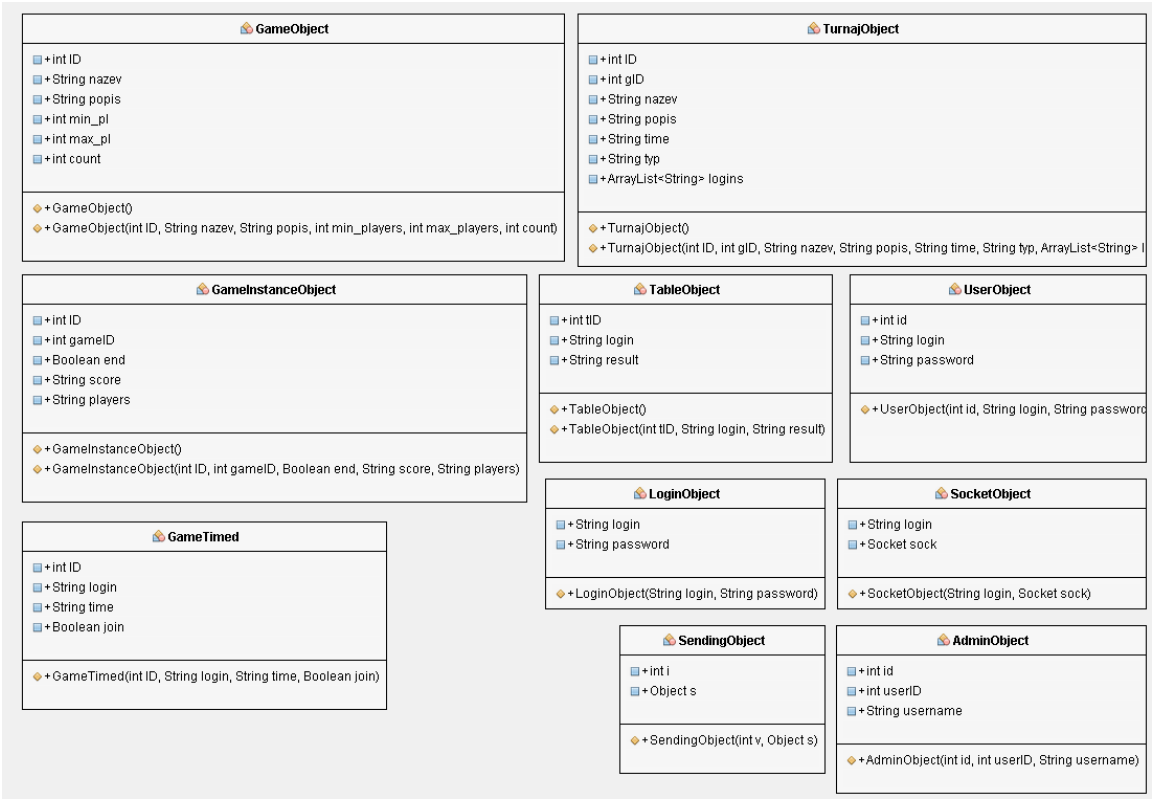
5 Popis implementace

Projekt je rozdělen na 3 programy a funguje na principu server-client. Server zajišťuje síťovou komunikaci mezi uživateli a spojení s databází. Client a Web obsahují uživatelské rozhraní, přes které uživatel bude zasílat požadavky na server.

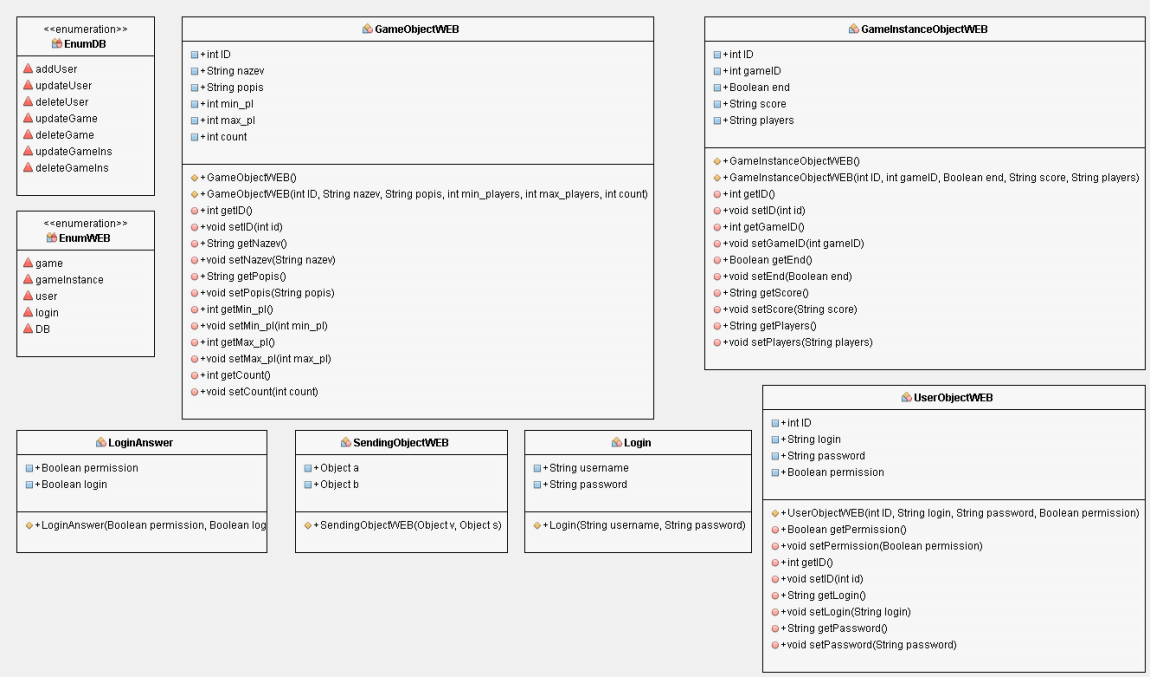


Obrázek 12: zjednodušené schéma programu

Projekt obsahuje balíčky object a objectWEB, které obsahují pomocné třídy. Hlavní využití těchto tříd je na zapouzdření dat před síťovým přenosem.



Obrázek 13: schéma tříd v baličku object

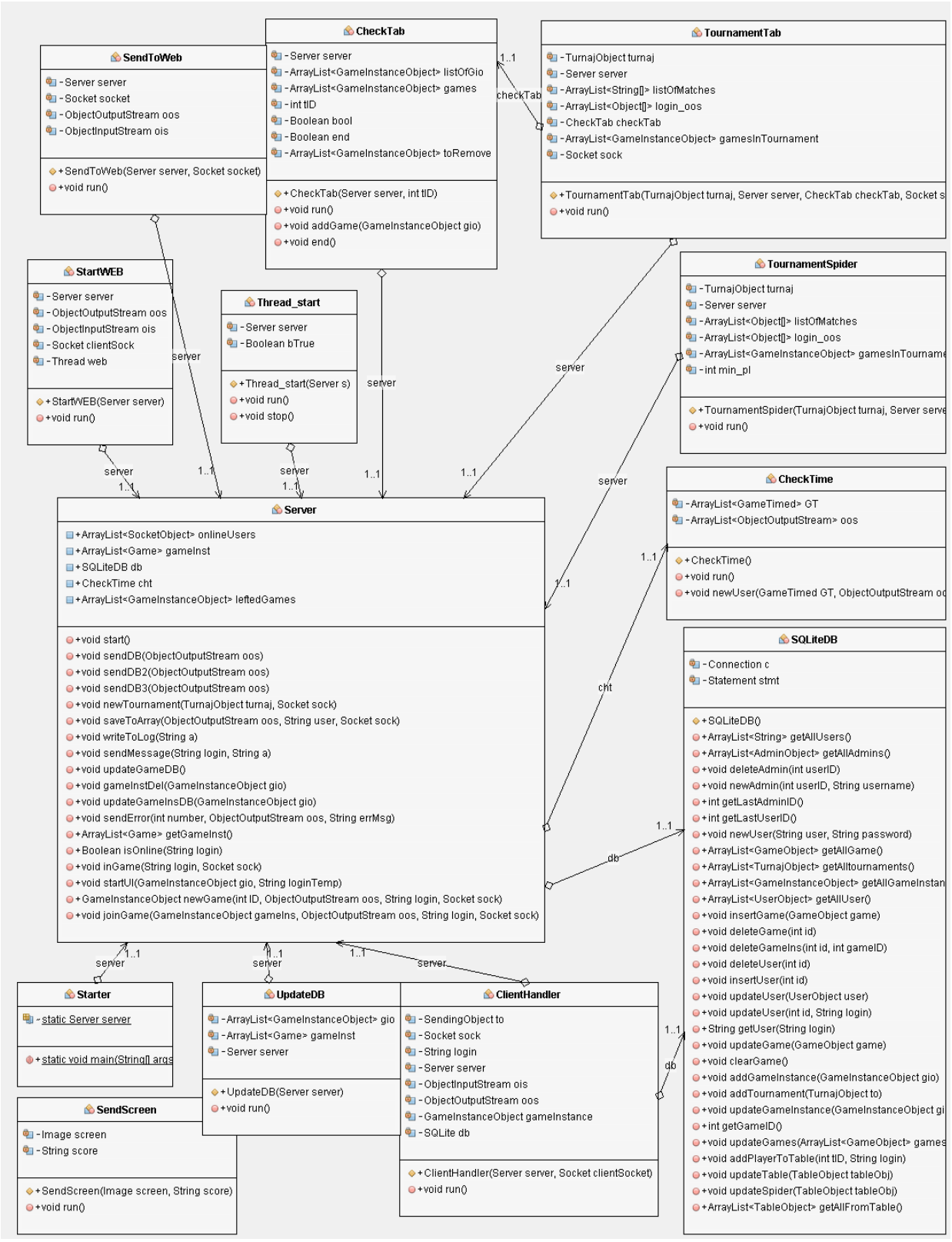


Obrázek 14: schéma tříd v balíčku objectWEB

5.1 Server

Veškerá síťová komunikace, která se nachází v projektu je řešena v programu Server. Většina tříd je řešena pomocí vláken, aby mohl server obsluhovat více uživatelů najednou. Server obsahuje několik tříd, které jsou rozděleny do čtyř balíčků.

- game - obsahuje Interface pro hry a jeho dědice
- object - obsahuje pomocné třídy pro síťovou komunikaci s programem Client, schéma tříd tohoto balíčku je na obrázku 13
- objectWEB - obsahuje pomocné třídy pro síťovou komunikaci s webovou aplikaci, schéma tříd tohoto balíčku je na obrázku 14
- server - obsahuje třídy, které zajišťují funkcionalitu programu, schéma tříd tohoto balíčku je na obrázku 15



Obrázek 15: schéma tříd v baličku server

Hlavní třídou je Starter, který vytvoří novou instanci třídy Server. Při vytvoření instance Server se vytvoří několik instancí tříd, které fungují jako vlákna a běží na pozadí serveru.

Třída server obsahuje několik seznamů například seznam aktivních hráčů a seznam aktuálně probíhajících her. Tyto seznamy jsou uloženy v kolekcích ArrayList. Dále přes třídu přistupujeme do instance SQLite, která obsluhuje databázi. Třída taky obsahuje několik metod, které využívají jiné třídy například metoda newGame(), která vytváří novou hru a je využívaná třídami ClientHandler, TournamentSpider a TournamentTab.

SQLiteDB je třída, která obsluhuje databázi. Podrobněji ji budu rozebírat v kapitole 5.4 Databáze.

Další třídou je Thread_Start, která je spuštěná na novém vlákně a zachytává nové pokusy o připojení na server. Pokud zachytí nové připojení, tak vytvoří novou instanci třídy ClientHandler.

Každá instance třídy ClientHandler obsahuje jednoho uživatele. Hlavní funkce této třídy je přijímání síťové komunikace od programu Client a následné zpracování dat ze síťového spojení. Data zpracovává pomocí prvku switch. Všechny objekty, které se přeposílají přes síť jsou instance třídy SendingObject. Každá instance SendingObject má dvě proměnné, první z nich je datového typu int a podle jeho hodnoty zjišťujeme jaký je druhý objekt a zároveň co uživatel chce udělat. Například pokud přijde číslo 0 tak víme, že uživatel se chce přihlásit a druhý objekt v SendingObject je LoginObject, který obsahuje přihlašovací jméno a heslo.

Třída UpdateDB provádí aktualizaci databáze. V celé aplikaci se vytvoří jen jedna instance této třídy a vytvoří se ihned po zapnutí aplikace. Instance je spuštěná jako vlákno a každých 5 vteřin projde seznam nedohraných her a zjišťuje zda-li už nějaké hry neskončili. Pokud nalezne již dohrané hry v seznamu nedohraných her, tak aktualizuje data v databázi a hru vymaže ze seznamu.

TournamentSpider je třída, ve které se vytváří turnaje typu pavouk. Pro každý turnaj se vytvoří nová instance v novém vlákně. Metoda run() začíná cyklem, který kontroluje čas a porovnává ho se začátkem turnaje. V případě, že se čas shoduje začne program zjišťovat kdo z hráčů není aktivní a vymaže ho ze seznamu hráčů a cyklus se ukončí. Následuje další cyklus, který je ukončen v případě že je turnaj dokončen. V cyklu vytvoříme zápasy a do zápasů náhodně vložíme hráče a vytvoříme nové instance her. Poslední část této metody je čekáni na výsledky. Výherce si zapíšeme do seznamu a vrátíme se na začátek cyklu, kde se vytvoří nové zápasy jen s výherci. Pokud nějaký hráč přebývá a zůstal by po přiřazení hráčů do her sám, tak je vybrán jeden náhodný hráč a ten automaticky postoupí do dalšího kola.

TournamentTab funguje podobně jako TournamentSpider, akorát se v metodě run() nečeká na výsledky. Výsledky turnaje se následně kontrolují a ukladají do databáze ve třídě CheckTab.

Třida CheckTime kontroluje seznam s naplánovanými zápasy. Když se čas zápasu shoduje s aktuálním časem zahájí se síťová komunikace s Clientem a zašle se zpráva s žádostí o zahájení zápasu.

Ve třídě SendScreen se nachází metoda run(), která vyžádá skóre a obrázek z probíhající hry a přepošle obrázek přes síťovou komunikaci.

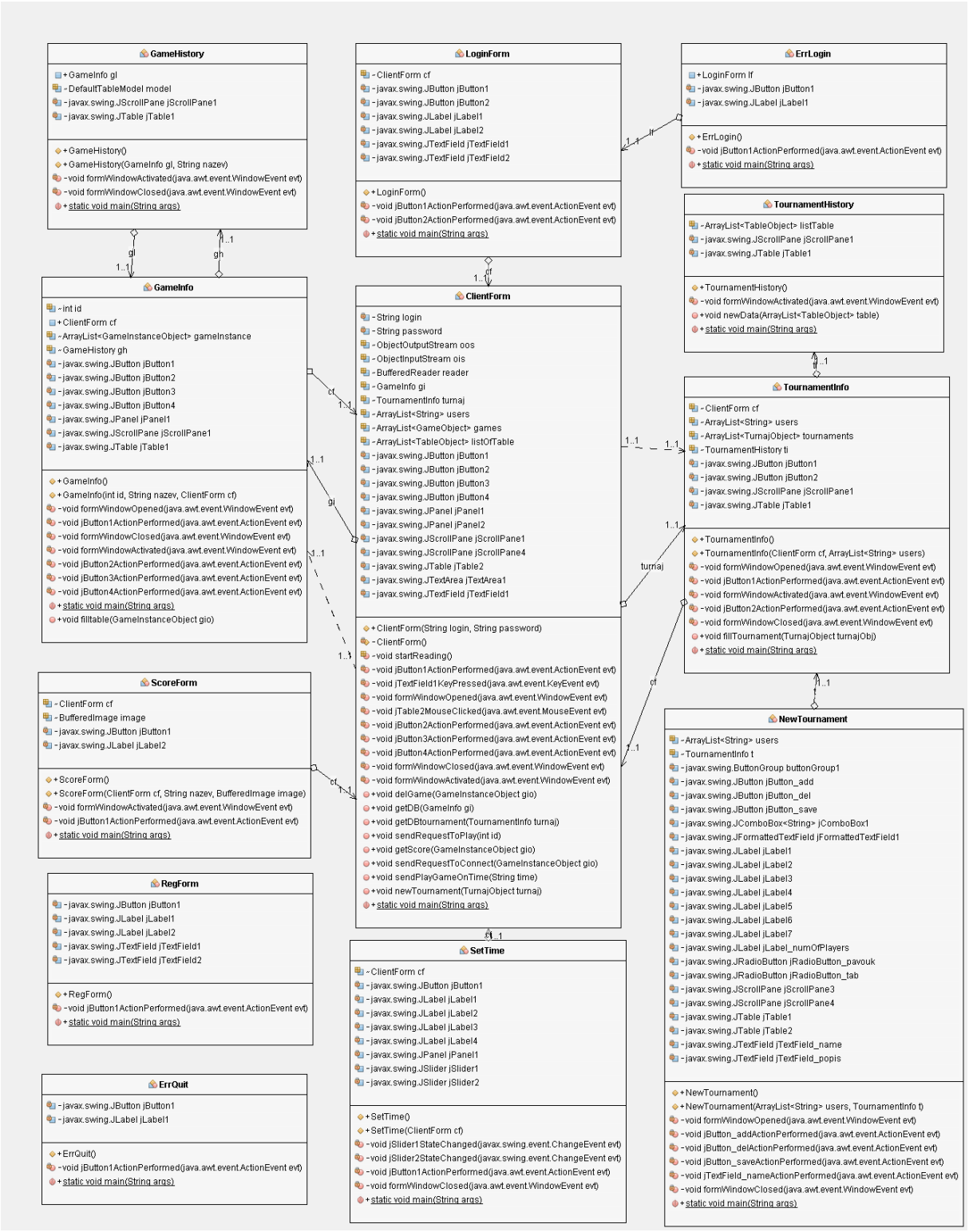
StartWeb obsahuje metodu run(), která se spustí vytvoření instance serveru. V metodě run() je nekonečný cyklus. V cyklu se čeká na nový požadavek od webové stránky a při přijmutí nového požadavku se vytvoří nová instance třídy SendToWeb.

Třída SendToWeb je podobná třídě ClientHandler. Rozdíl mezi těmito třídami je ten, že ClientHandler obsluhuje přes síťové spojení program Client a třída SendToWeb obsluhuje přes síťové spojení webovou aplikaci. Nejdůležitější prvek v celé třídě je switch, který pomocí EnumWEB třídí a zpracovává dotazy z webové stránky.

5.2 Client

Program Client obsahuje grafické uživatelské rozhraní, ukázky rozhraní můžete vidět na obrázcích 17 a 18. Běžný uživatel bude pomocí tohoto rozhraní ovládat celou aplikaci. Bude mít možnost vytvářet nové hry, plánovat hry na určitý čas, vytvářet nové turnaje, zobrazovat si odehrané turnaje a další. Client obsahuje několik tříd, které jsou rozděleny do tří balíčků.

- game - obsahuje grafické uživatelské rozhraní her
- object - obsahuje pomocné třídy pro síťovou komunikaci se serverem, schéma tříd tohoto balíčku je na obrázku 13
- client - obsahuje třídy, které obsahují grafické rozhraní, schéma tříd tohoto baličku je na obrázku 16

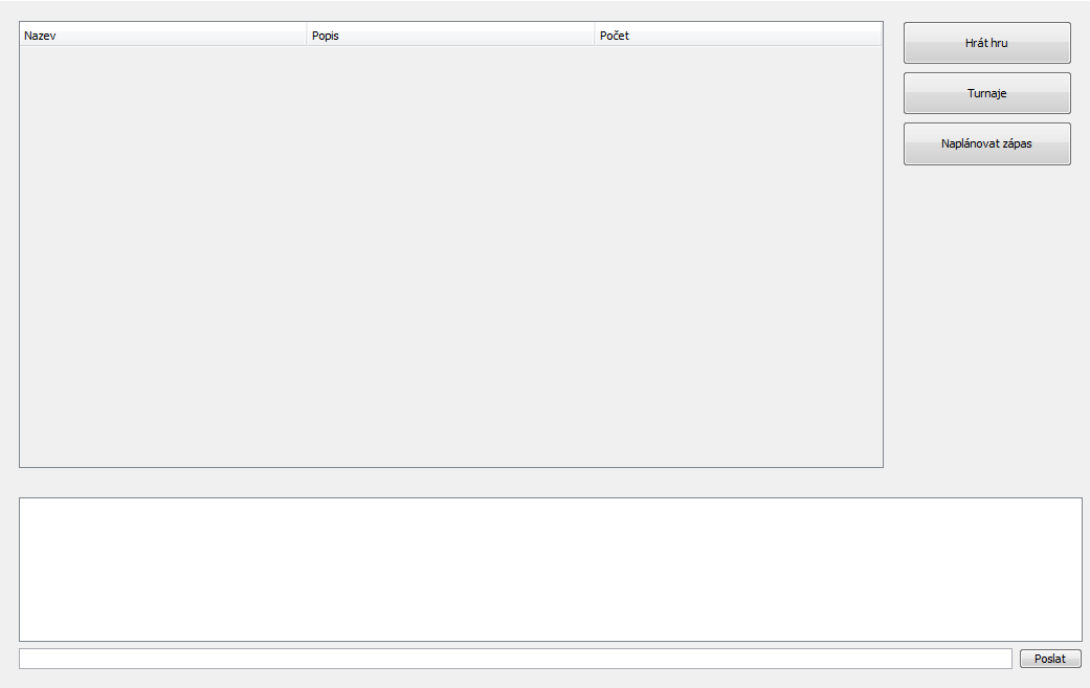


Obrázek 16: schéma tříd balíčku client

Hlavní třídou je LoginForm, který má pouze dvě funkce. První funkce je přihlásit a druhá funkce je registrovat. Při registraci se vytvoří instance třídy RegForm. Při přihlášení se vytvoří nová instance třídy ClientForm.

RegForm má jen jednu funkci a to zaslat přihlašovací údaje na server, aby si je zapsal do databáze a tím uživatele registroval.

Třída ClientForm zachycuje veškerou síťovou komunikaci od serveru. Přesněji řečeno třída ClientForm obsahuje vnořenou třídu IncomingReader, která zachytává síťovou komunikaci a zachycené data zpracovává. Vše je řešeno podobně jako v programu Server, tedy pomocí prvku switch, který třídí požadavky podle hodnoty v SendingObjectu. ClientForm dále obsahuje několik metod, které posílají žádosti na server. Například žádost o databázi her nebo žádost o spuštění hry. Při otevření uživatelského rozhraní se provede metoda formWindowOpened, která nejdříve ověří přihlašovací údaje a potom zašle žádost o seznam her, které se mají zobrazit v uživatelském rozhraní, které můžeme vidět na obrázku 17.



Obrázek 17: uživatelské rozhraní - CleintForm

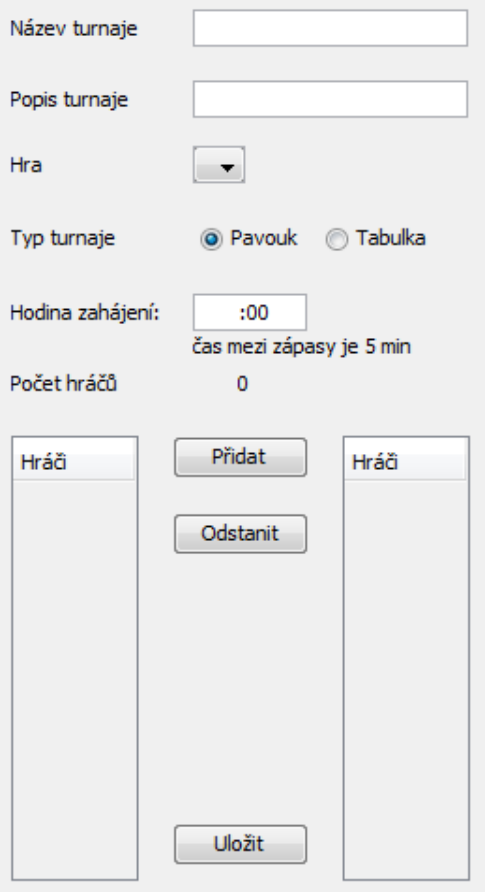
GameInfo v grafickém rozhraní zobrazí všechny dostupné hry, které ještě nebyli ukončeny. Na tyto hry se lze připojit a nebo zobrazit aktuální skóre. Další funkcí je vytvořit novou hru, což se provede zavoláním stisknutím tlačítka s nápisem "nová hra". Stisknutí zavolá metodu jButton1ActionPerformed, která zavolá metodu sendRequestToPlay, která je uložena ve třídě ClientForm.

GameHistory zobrazuje v grafickém rozhraní všechny již odehrané zápasy.

ScoreForm je třída pro zobrazení aktuálního průběhu hry.

TournamentInfo je třída, která zobrazuje všechny turnaje a dává možnost uživateli přejít na vytvoření nového turnaje nebo zobrazení podrobností.

NewTournament je formulář pro vytvoření nového turnaje, formulář můžete vidět na obrázku 18. Při potvrzení se zavolá metoda newTurnament, která je uložena ve třídě ClientForm.



Obrázek 18: uživatelské rozhraní - NewTournament

Pro naplánování hry na určitý čas využijeme třídu SetTime. V grafickém rozhraní této třídy si vybereme čas, kdy má zvolená hra být spuštěna. Po potvrzení času se zavolá metoda s názvem sendPlayGameOnTime(), která je uložena v ClientForm.

Třídy ErrLogin a ErrQuit jsou třídy, které se provedou jen v případě chyby. ErrLogin se provede při chybných přihlašovacích údajích. Třída oznámí uživateli chybu a provede přesměrování zpět na LoginForm. ErrQuit se provede při situaci, kdy se Client nemůže připojit na Server a proto po oznámení uživateli o chybě se aplikace ukončí.

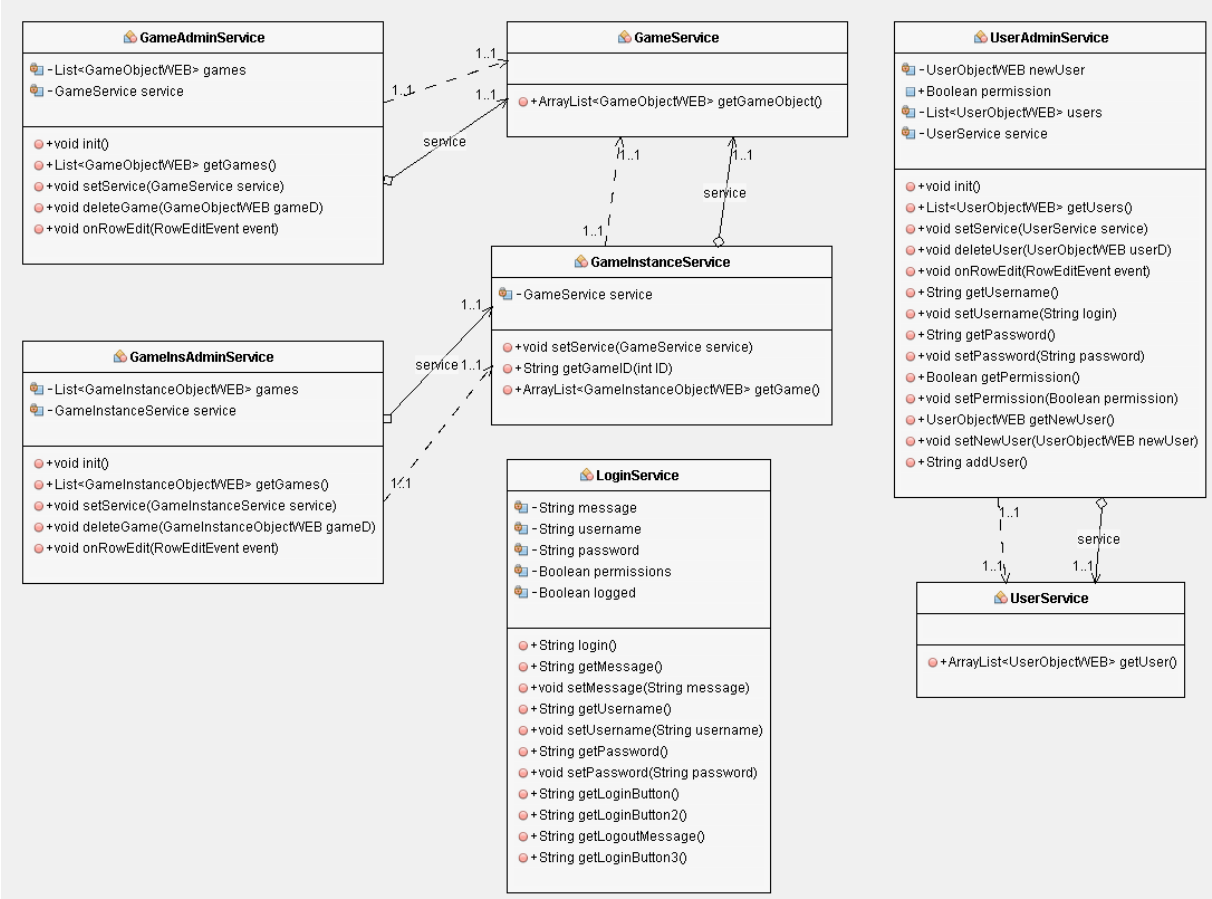
5.3 Webová aplikace

Pro vytvoření webové aplikace jsem využil technologii Java Server Faces a je použita knihovna PrimeFaces 5.0, která mimo jiné přidává komponenty do grafického uživatelského rozhraní. Celý web běží na serveru glassFish.

Webová aplikace obsahuje několik xhtml a java souborů. Soubory s příponou .xhtml definují vzhled stránky, který je možný vidět na obrázku 20. Funkcionalitu stránky zajišťují Beany, které

se nachází v souborech s příponou .java. Beany jsou normální třídy, které upravují hodnoty proměných v xhtml. souborech. Tyto java soubory jsou rozdělené do dvou baličků.

- objectWEB - obsahuje pomocné třídy pro síťovou komunikaci se serverem, schéma tříd tohoto balíčku je na obrázku 14
- service - obsahuje třídy, které zajišťují funkcionalitu webové aplikace, schéma tříd tohoto baličku je na obrázku 19



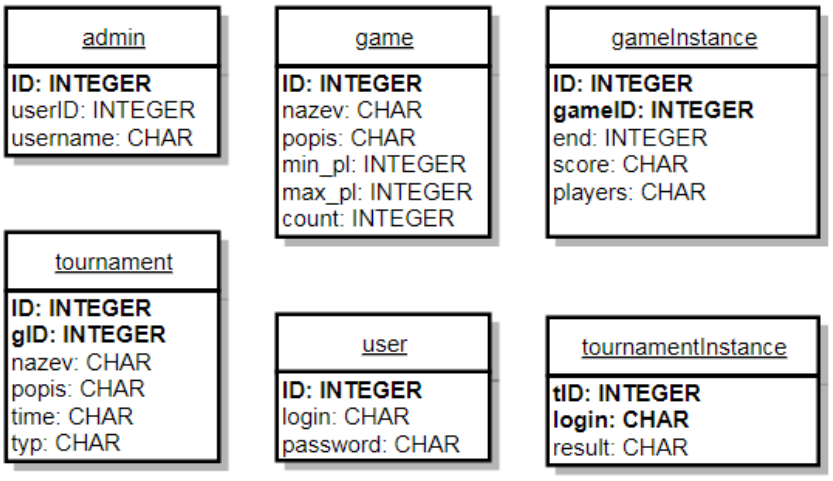
Obrázek 19: schéma tříd balíčku service

Hlavní funkcí webu je správa her a uživatelů. Pro možnost upravit hráče nebo hry je nutnost se přihlásit a mít dostatečnou pravomoc. Pro komunikaci se serverem je využité síťové spojení. Samotná webová aplikace nemá přistup k databázi vše se tedy musí řešit přes server.

Obrázek 20: webová stránka

5.4 Databáze

Databáze v tomto programu slouží jako záloha dat. S databází pracuje třída SQLiteDB, která využívá knihovnu SQLiteDriver. Přístup k databázi je řešen přes JDBC. Třídá SQLiteDB obsahuje konstruktor ve kterém se vytvoří připojení k databázi SQLite. Instance této třídy je v celém serveru jen jedna. Pro každý různý SQL příkaz, který se má provést na databázi je ve třídě jedna metoda.



Obrázek 21: jednoduché schéma entit databáze

Databáze obsahuje šest tabulek.

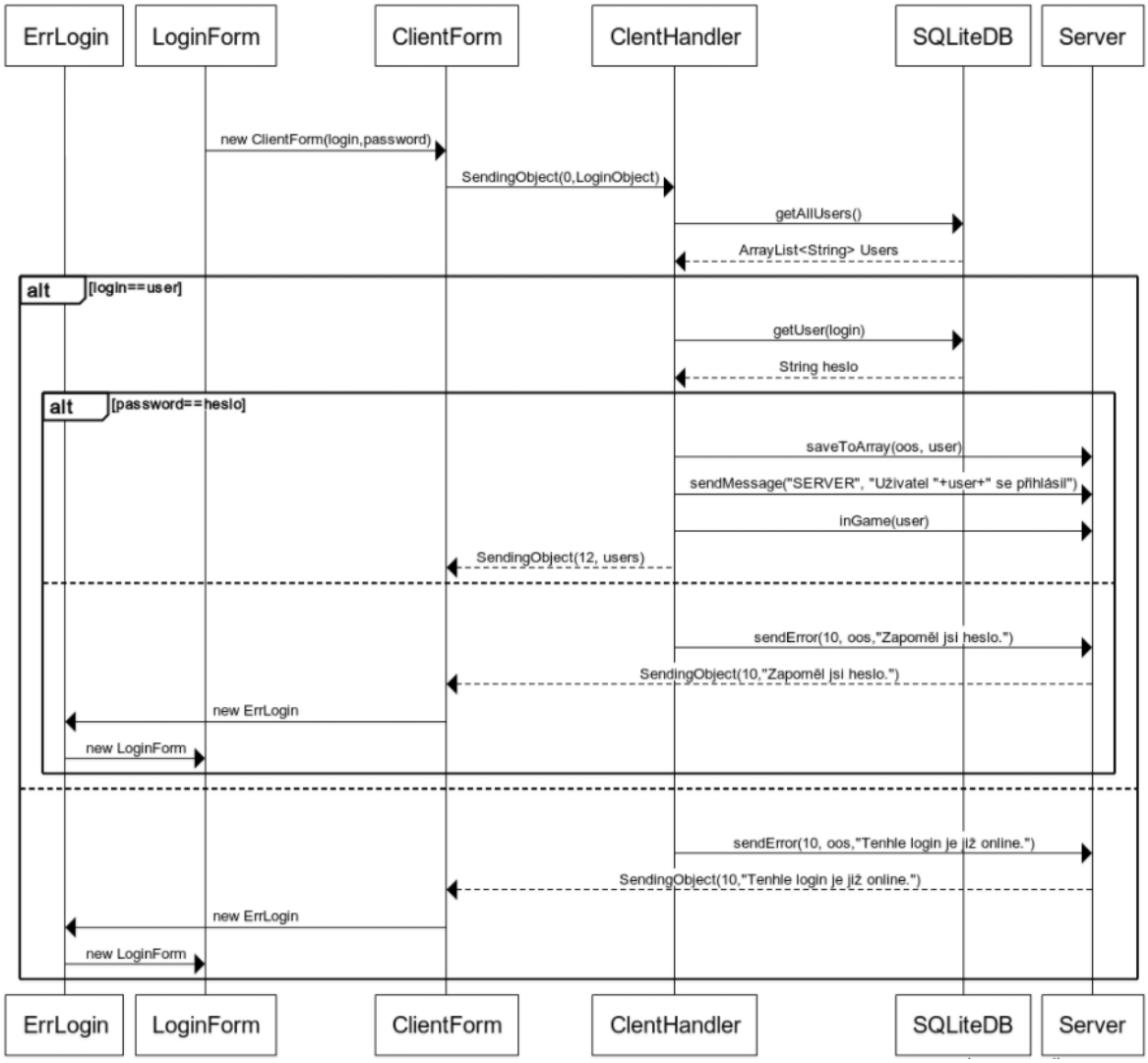
- admin
  - pokud se uživatel nachází v této tabulce tak má práva na editaci záznamů
  - userID je cizí klíč
  - primárním klíčem je ID
- game
  - obsahuje seznam všech dostupných her
  - primárním klíčem je ID
- gameInstance
  - obsahuje všechny probíhající i odehrané hry
  - gameID je cizí klíč
  - primárním klíčem je kombinace ID a gameID
- tournament
  - obsahuje všechny dostupné turnaje
  - gID je cizí klíč
  - primárním klíčem je kombinace ID a gID
- user
  - obsahuje základní údaje o uživateli
  - primárním klíčem je ID
- tournamentInstance
  - obsahuje výsledky zápasů z turnajů
  - tID a login jsou cizí i primární klíče

5.5 Popis funkcí

5.5.1 Přihlášení na server

Třídy ErrLogin, LoginForm a ClientForm jsou součástí programu Client. Zbývající třídy patří do programu Server. Přihlášení začíná v LoginForm, kde uživatel zadá přihlašovací údaje a při potvrzení se vytvoří instance třídy ClientForm, která má jako parametry přihlašovací údaje. ClientForm v metodě formWindowOpened se přihlašovací údaje se vloží do třídy LoginObject. Instance třídy loginObject se vloží do SendingObject a přidá se číslo 0, aby server poznal že

se jedná o přihlášení. Třída ClientHandler zachytí zprávu, která obsahuje žádost o přihlášení a přihlašovací údaje a vyžádá si seznam uživatelů z databáze. Nyní se projde seznam uživatelů a porovná se s přihlašovacím jménem. V případě shody zažádá o heslo daného uživatele pomocí metody getUser a porovná heslo s heslem uživatele. V případě shody se na serveru provedou metody saveToArray,sendMessage, inGame a zašle se přes síťovou komunikaci zpráva o úspěšném přihlášení. V případě špatného hesla nebo nenalezeného uživatele se přes síťovou komunikaci zašle zpráva obsahující třídu SendingObject se zprávou o chybě. Ve třídě LoginForm se vytvoří nová instance třídy ErrLogin.

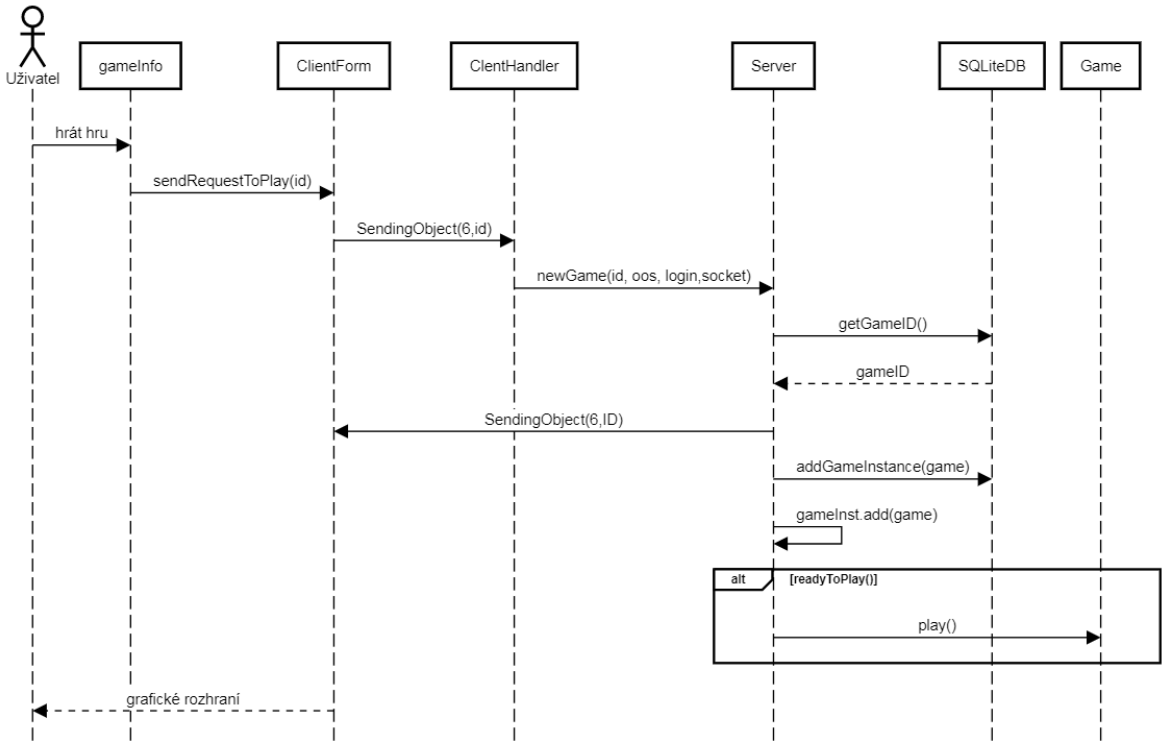


Obrázek 22: přihlášení na server

5.5.2 Zapnout hru

Uživatel zadá pokyn k vytvoření nové hry, pomocí tlačítka s nápisem "nová hra". Ve třídě GameInfo se zavolá metoda jButton1ActionPerformed, která zavolá na metodu sendRequestToPlay,

která se nachází v ClientForm. Vytvoří se nová instance třídy SendingObjekt s hodnotami 6 a ID hry. Tato instance se zašle přes síťové spojení na server, kde zprávu zachytí třída ClientHandler, která zjistí dle čísla 6, že se jedná o založení nové hry a zavolá metodu newGame, která se nachází v třídě server. V metodě newGame se zjistí jaké bude mít nová hra ID potom se zjistí, která hra se má vytvořit a následně se vytvoří nová instance hry. Přes síťové spojení se odešle SendingObject, který bude obsahovat číslo 6 a ID hry. Hra se přidá do databáze a do seznamu,který se nachází v třídě server. Následně server pošle dotaz na hru zda-li není již připravená ke hraní. Potom se buď to hra spustí nebo uživatel bude čekat na dalšího hráče. Mezitím třída ClientForm zachytila zprávu od serveru a vytvoří grafické rozhraní pro danou hru.



Obrázek 23: zapnout hru

5.5.3 Zapnutí umělé inteligence

Při výskytu výjimky, v přijímání zpráv ze síťového spojení, se ve třídě ClientHandler provede podmínka zda-li je připojení stále aktivní. Pokud připojení aktivní není tak se zjistí jestli byl hráč v nějaké hře a pokud ano zavolá se metoda startUI na hru, aby nahradila hráče umělou inteligencí. V případě, že umělá inteligence nahradila hráče tak se do databáze do výsledků zapíše před uživatelské jméno "AI-".

6 Závěr

Cílem této bakalářské práce bylo vytvořit herní portál v jazyce Java, který by umožňoval hraní her přes internet. Portál měl umožňovat například správu uživatelů a her,vytváření turnajů, zakládání nových her a další. Herní portál měl obsahovat desktopovou a webovou aplikaci.

Začátek této práce byl věnován teorii okolo jazyka Java. Dále jsem se v této práci věnoval technologiím jazyka Java, které byli v aplikaci využity. Následně jsem popsal implementaci programu, kterou jsem doplnil o několik diagramů. Závěrem téhle práce jsem se věnoval popisováním několika funkcí aplikace, které byli opět doplněny o diagramy.

Herní portál je funkční a splňuje většinu požadavků. V práci není splněn bod, který požadoval spouštění hry jako Applet. Tento bod není splněn z důvodu, že Applet přestal být podporován většinou webových prohlížečů.

V projektu není brán ohled na bezpečnost a proto by uvedení tohoto projektu do praxe vyžadovalo několik úprav.

Osobní přínos této práce bylo poznání vývoje většího projektu. A rozšíření znalostí v oboru programování.

Literatura

[1] Wikipedia: the free encyclopedia [online]. San Francisco (CA):Wikimedia Foundation, 2016 [cit. 2018-04-26]. Dostupné z: <https://cs.wikipedia.org/wiki/JDK>

[2] Wikipedia: the free encyclopedia [online]. San Francisco (CA):Wikimedia Foundation, 2016 [cit. 2018-04-26]. Dostupné z: <https://en.wikipedia.org/wiki/Write_once,_run_anywhere>

[3] Java VirualMachine JVM tutorial [online]. [cit. 2018-04-26]. Dostupné z: <http://viralpatel.net/blogs/java-virtual-machine-an-inside-story/>

[4] Zapouzdření. IT Netowrk. [online]. [cit. 2018-04-26]. Dostupné z: <https://www.itnetwork.cz/php/oop/tutorial-php-zapouzdreni-objektove-orientovane-programovani>

[5] SCHILDT, Herbert. Java 7: výukový kurz. 1. vyd. Brno: ComputerPress, 2012, 664 s. ISBN 978-80-251-3748-2.

[6] GUI. IT-slovnik. [online]. [cit. 2018-04-26]. Dostupné z: <https://it-slovnik.cz/pojem/gui>

[7] Understanding Streams in Java [online]. [cit. 2018-04-26]. Dostupné z: <https://msaudi.wordpress.com/page/3/>

[8] Webové aplikace - FI WIKI[online]. [cit. 2018-04-26]. Dostupné z: <https://kore.fi.muni.cz/wiki/index.php/Webov%C3%A9_aplikace>

[9] Java Server Pages - FI WIKI[online]. [cit. 2018-04-26]. Dostupné z: <https://kore.fi.muni.cz/wiki/index.php?title=Java_Server_Pages>

[10] Vývoj webových aplikací pomocí frameworku JavaServer Faces [online]. [cit. 2018-04-26]. Dostupné z: <https://java.vse.cz/jsf/chunks/ch07.html>

[11] GILFILLAN, Ian. Myslíme v MySQL 4: knihovna programátora. 1. vyd. Praha: Grada Publishing, 2003, 750 s. ISBN 80-247-0661-X.

[12] Relační databáze. Gymnázium Čakovice. [online]. [cit. 2018-04-26]. Dostupné z: <http://vyuka.greendot.cz/materialy/material-4.pdf>

[13] Hierarchická databáze. Česká terminologická databáze knihovnictví a informační vědy [online]. [cit. 2018-04-26]. Dostupné z: <http://aleph.nkp.cz/publ/ktd/00000/01/000000103.htm>

[14] Síťová databáze. Česká terminologická databáze knihovnictví a informační vědy [online]. [cit. 2018-04-26]. Dostupné z: <http://aleph.nkp.cz/publ/ktd/00000/01/000000128.htm>

[15] Úvod do JDBC. Interval [online]. [cit. 2018-04-26]. Dostupné z: <https://www.interval.cz/clanky/uvod-do-jdbc/>

[16] JDBC - Driver Types. Tutorial points. [online]. [cit. 2018-04-26]. Dostupné z: <https://www.tutorialspoint.com/jdbc/jdbc-driver-types.htm>

Seznam příloh

Součástí bakalářské práce je DVD, které obsahuje:

1. Bakalářská práce.pdf - vlastní práce
2. Project - Projekt herní portál java